UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/722,183	11/24/2003	Robert Stanley Kolman	10030573-1	7018

63448 7590 01/07/2009

Gregory W. Osterloth
Holland & Hart, LLP
P.O. Box 8749
Denver, CO 80201

EXAMINER	
LE, TOAN M	
ART UNIT	PAPER NUMBER
2863	

MAIL DATE	DELIVERY MODE
01/07/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/722,183
Filing Date: November 24, 2003
Appellant(s): KOLMAN ET AL.

Gregory W. Osterloth
For Appellant

EXAMINER'S ANSWER

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

This is in response to the appeal brief filed 10/17/08 appealing from the Office action mailed 12/28/07.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-7, 10-16, and 19-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Colby et al. (US Patent No. 6,622,271) and further in view of Gygi et al. (US Pub No. 2003/0235156 A1).

Referring to claim 1, Colby et al. disclose an apparatus, comprising:

computer readable media; and

program code, stored on the computer readable media (figures 1A and 1B), comprising:

code to define a user interface 72 (figure 1A) (col. 4, lines 41-48);

code to detect invalid test definition data in user input (col. 4, lines 54-67 to col. 5, lines 1-4; col. 11, lines 45-57; col. 12, lines 20-29); and

code to receive a valid data option selected through the user interface, and to update the invalid test definition data with the valid data option (col. 11, lines 52-57).

As to claim 6, Colby et al. disclose an apparatus, wherein at least some of said user input is received through said user interface (figures 1A and 1B).

Referring to claim 7, Colby et al. disclose an apparatus, wherein at least some of said user input is contained in a test definition file (col. 6, lines 19-39; col. 11, lines 58-67 to col. 12, lines 1-2).

Referring to claim 11, Colby et al. disclose an apparatus, wherein the user interface comprises code to define an input area to receive a specification for invalid test definition data that has been detected as invalid because it lacks a specification to make it valid (col. 12, lines 20-29).

As to claim 12, Colby et al. disclose an apparatus, wherein said input area to receive a specification for invalid test definition data is configured to receive a data type (col. 12, lines 20-29).

As to claim 14, Colby et al. disclose a computer-based method, comprising:

parsing user input to detect invalid test definition data in the user input (col. 4, lines 54-67 to col. 5, lines 1-4; col. 11, lines 45-57; col. 12, lines 20-29);

upon receiving a valid data option selected from the set of valid data options, updating the invalid test definition data with the valid data option (col. 11, lines 55-57); and

generating circuit test data structures to control an automated circuit tester (figures 1A, 1B, 4-5).

Referring to claim 15, Colby et al. disclose a computer-based method, wherein parsing user input comprises parsing a test definition file (col. 6, lines 19-39; col. 11, lines 58-67 to col. 12, lines 1-2).

As to claim 16, Colby et al. disclose a computer-based method, further comprising compiling the set of valid data options based on a context of the invalid data (col. 5, lines 44-48).

As to claim 19, Colby et al. disclose a computer-based method, comprising:

parsing source code for generating circuit test data structures, to identify type name definitions and enumeration constant definitions contained in said source code (figures 4-5; col. 10, lines 34-41);

generating a string table from said type name and enumeration constant definitions (figures 4-5; col. 10, lines 34-41); and

linking said string table to an input validation and error messaging portion of said source code to i) cause said source code to index said string table upon detection of invalid test definition data in user input (col. 10, lines 22-41).

Referring to claim 20, Colby et al. disclose a computer-based method, wherein said index into said string table comprises a context of said invalid test definition data (col. 5, lines 44-48).

Colby et al. do not teach upon detection of invalid test definition data, prompt a user to select a valid data option from a set of valid data option, said prompting being undertaken through the user interface, code to compile the set of valid data options based on a context of the invalid test definition data *as in claim 2* to index a table of valid data options as in claim 3, to parse the user input and log valid data options into the table as in claim 4, wherein the context comprises a data type as in claim 5, the code to configure how the set of valid data options is displayed through the user interface as in claim 10, and the set of valid data options comprises a single valid data option that is replaceable by the user as in claim 13, or cause a set of valid data options retrieved from the string table to be displayed to a user for user selection as in claim 19.

Gygi et al. disclose an apparatus, comprising :

computer readable media; and

program code, stored on the computer readable media, comprising:

code to define a user interface;

code to detect invalid test definition data in user input and, upon detection of invalid test definition data, prompt a user to select a valid data option from a set of valid data option, said prompting being undertaken through the user interface, *code to compile the set of valid data options based on a context of the invalid test definition data* to index a table of valid data options, to parse the user input and log valid data options into the table, wherein the context comprises a data type, the code to configure how the set of valid data options is displayed through the user interface, and the set of valid data options comprises a single valid data option that is replaceable by the user, and cause a set of valid data options retrieved from the string table to be displayed to a user for user selection ([0048], [0050], [0051], [0068], and [0069].

Accordingly, it would have been obvious to one having ordinary skill in the art at the time the invention was made to have applied the teaching of Gygi et al. into the reference of Colby et al. to assist automated testing systems through standardized user interface and programming interface for performing circuit tests.

Allowable Subject Matter

Claims 8-9 and 17-18 are objected to as being dependent upon a rejected base claims 1 and 14, respectively, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

The reason for allowance of claims 8-9 and 17-18 is the inclusion of the code that prompts a user to select a valid data option causes the set of valid data options to be displayed through the user interface in alphabetical order and in order of highest likelihood of correctness.

(10) Response to Argument

Referring to claim 1, Appellant argues:

"A user cannot select a valid data option through a user interface unless the user is first prompted with some sort of option to select. However, as admitted by the Examiner, Colby does not disclose any sort of prompting of a user 'to select a valid data option'. See, 12/28/07 Final Office Action, pp. 2-3. Appellants therefore assert that Colby does not disclose 'code to receive a valid data option selected through [a] user interface'. Appellants do admit, however, that Colby discloses an interface to receive user input (but an interface that merely receives an input thought of by a user is not an interface that provides an option for a user to select).

Turning now to the recitation of claim 1 that recites, 'code to …, upon detection of invalid test definition data, prompt a user to select a valid data option from a set of valid data options; said prompting being undertaken through the user interface', the examiner admits that this is not taught by Colby. However, the Examiner asserts that this is taught by Gygi in, for example, paragraphs [0048], [0050], [0051], [0068] and [0069]. Appellants respectfully disagree.

In paragraph [0051], Gygi appears to teach prompting a user to select a valid option from a set of valid data options. Gygi also indicates that a parameter definition language enables a test designer to specify ranges, permissible values, and messages that help a user to select a test parameter. Of note, however, Gygi says nothing about 'code to detect invalid test definition data' or 'code to …, upon detection of invalid test definition data, prompt a user to select a valid data option'. Instead, Gygi discloses how a parameter definition language may be used to influence or limit a user's input, so that invalid test definition is never provided by the user.

Having summarized the teachings of Colby and Gygi individually, Appellants turn now to whether it would have been obvious to one of ordinary skill in the art, at the time of Appellants' invention, to combine the teachings of Colby and Gygi. The Examiner asserts that it would have been obvious 'to have applied the teaching of Gygi et al. into the reference of Colby et al. to assist automated testing systems through standardized user interface and programming interface for performing circuit tests.' See, 12/28/2007 Final Office Action, p. 3. Appellants disagree.

On one hand, Colby discloses a system for detecting invalid test definition data and prompting a user to supply valid test definition data. However, Colby's system does not give a user any guidance on what kind of test definition data might be valid, and instead relies on the user to know what test definition data would be valid. On the other hand, Gygi discloses a system that provides a user with so much guidance and structure that receiving (or needing to detect) invalid test definition data is eliminated. If Gygi's parameter definition language and associated user interface were to be incorporated into Colby's system, there would no longer be a need for Colby's system to 'detect invalid test definition data', because invalid test definition data would presumably be prohibited by Gygi's structured parameter input interface. Given this dichotomy resulting from 1) Gygi's invention being 'preventive' of invalid test definition data, and 2) Colby's invention being 'curative', Appellants do not believe that one of ordinary skill in the art, at the time of Appellants' invention, would have found it obvious to combine (and modify) the disclosures of Colby and Gygi to achieve the apparatus set forth in Appellants' claim 1.

Answer:

Colby teaches code to detect invalid test definition data in user input (col. 4, lines 54-67 to col. 5, lines 1-4 and col. 11, lines 45-57 and col. 12, lines 20-29) such as:

"The processor 26 also executes a tester rules checker program 76, which includes portions 77 and 78 that each corresponds to a respective one of the testers 17 and 18. The rules checker program 76 evaluates the test definition 73 generated by the test definition generator 71, in order to verify that the test definition 73 is compatible with the capabilities of each of the testers on which it is to be used, including the testers 17 and 18." (col. 4, lines 54-8)

"If the rules checker program 76 detects a problem, it will provide a message at 79 to the test definition generator 71, so that a warning message can be presented to the operator." (col. 5, lines 1-4)

"If a problem is detected, then a warning message is provided to the operator, so that appropriate adjustments can be made to the test definition 73." (col. 11, lines 52-55)

"If any errors are detected, the interpreter program 131 will provide the operator with an identification of those errors. The operator has the capability to carry out certain debug functions of a standard type, such as setting breakpoints, dynamically changing the values of variables, and so forth. Further, the operator can instruct the interpreter program 131 to maker changes to the modified test definition 301, for example to correct errors which were present in the initial test

definition 73, or to implement special test conditions to help identify an elusive problem in a particular device 12." (col. 12, lines 20-29)

Thus, Colby does disclose code to detect valid and invalid test definition data in user input.

Even though Colby does not specifically mention "upon detection of an invalid test definition data, prompt a user to select a valid data option from a set of valid data options, said prompting being undertaken through the user interface." (emphasis by underline)

Gygi discloses "The invention also includes a flexible command, status and parameter definition language that permits a test designer to define a wide variety of custom commands, test specific status and test parameters to be supplied by the test operator in starting a selected test vehicle. **Parameters** of test vehicles may be defined in the definition language. Status information unique to a particular test may also be defined by the language. In addition, entire custom commands may be defined by the language. *The definitions includes types and ranges of permissible values as well as user interface information to* ***prompt*** *the test operator for desired values*. (paragraph [0048])

Gygi also discloses "This flexible language is provided as a portion of the test vehicle and defines the user interface to be provided to the test operator to supply requisite parameters to start a test. The command language includes a number of keywords or directive to define the

nature of each parameter to be supplied to the test vehicle on startup. *Fields of the keywords to define a parameter also include text used to present readable options to the test operator to select desired values for the various parameters of the test vehicle*." (paragraph [0050])

Gygi further discloses "The "parameter" keyword allows the test designer or operator to specify different parameters types and use them in the test or in custom tests (defined by the "command" keyword discussed below). Fields of the parameter keyword allow the parameter to be defined as a specific type such as a numeric value or a list of enumerated values for the user **to choose from** (i.e., *a list of options*). **Additional fields of the parameter keyword provide default values for the parameter to assume if the operator does not specify a value for the parameter when prompted to do so**. Still other fields allow a test designer to provide textual help messages to describe, for example, the usage of the parameter or permissible values for the parameter." (paragraph [0051]).

Thus, Gygi teaches upon detection of invalid test definition data, prompt a user to select a valid data option from a set of valid data options, said prompting being undertaken through the user interface such as "*The **definitions** includes **types** and **ranges of permissible values** as well as user interface information **to prompt** the test operator for **desired values.***" And "*Fields of the keywords to define a parameter also include text used to present readable **options** to the test operator to select desired values for the various parameters of the test vehicle.*" And "**Additional fields of the parameter keyword provide default values for the parameter to**

assume if the operator does not specify a value for the parameter when prompted to do so."

Thus, Colby teaches code to detect invalid test definition data in user input by providing the operator with an identification of those errors/invalid test definition.

And Gygi teaches upon detection of invalid test definition data, prompt a user to select a valid data option from a set of valid data options, said prompting being undertaken through the user interface by having definitions includes types and range of permissible values as well as user interface information to prompt the test operator for desired values including text used to present readable options to the test operator to select desired values for the various parameters of the test vehicle by defining specific type such as a numeric value or a list of enumerated values for the user to choose from (i.e. a list of options) so that it would have been obvious to one having ordinary skill in the art to combine the teaching of Colby's and Gygi's for improving a single test definition in a tester independent language with multiple different test systems.

In summary, Colby teaches code to detect both valid and invalid test definition data as indicated above, but Colby does not teach upon detection of invalid test definition data, prompt a user to select a valid data option from a set of valid data options. And Gygi discloses a teaching to overcome the deficiencies in the Colby's reference by providing code for prompting a user to select a valid data option from a set of valid data options upon detection of invalid test definition data so that the combination of Colby's and Gygi reference improves a single test definition tester independent language to provide a universal single test definition format to verify that the

test definition data is compatible with the capabilities among the multiple different tester systems.

Referring to claim 2, Appellant argues:

"With respect to Appellants' claim 2, the Examiner asserts that Gygi teaches 'code to compile the set of valid data options based on a context of the invalid test definition data' in one or more of paragraphs [0048], [0050], [0051], [0068] and [0069]. See, 12/28/2007 Final Office Action, p. 3. Appellants respectfully disagree.

Although Gygi discloses a parameter definition language that enables a test designer to both 1) define a test parameter, and 2) guide a user's input of a valid test definition parameter, Gygi does not disclose any means for detecting or otherwise identifying invalid test definition data. As a result, Gygi necessarily fails to disclose the compilation of a set of valid data options 'based on a context of...invalid test definition data', as recited in Appellants' claim 1."

Answer: Gygi teaches code to compile the set of valid data options based on a context of the invalid test definition data such as *the definitions includes types and ranges of permissible values as well as user interface information to prompt the test operator for desired values.* (paragraph [0048])

Thus, Gygi discloses the compilation of the set of valid data options based on a context of the invalid test definition data including types and ranges of permissible values/test definition data to prompt a test operator for desired values through use of user interface.

For the above reasons, it is believed that the rejections should be sustained.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

Respectfully submitted,

/Toan Le/

Toan Le
December 24, 2008

Conferees:

Drew A. Dunn
/D. A. D./
Supervisory Patent Examiner, Art Unit 2863

David Blum
(R.M. for David Blum)

Ricky Mack
/R. L. M./
Supervisory Patent Examiner, Art Unit 2873